

	Type	L #	Reqs	Search Text	DBs	Time Stamp
1	BRS	L1	5	(page or web or webpage or web-page or (web adj page)) near4 (weight\$5 and scor\$4 and relevan\$5)	USPAT; US-PG PUB; EPO; JPO; DERW ENT; IBM_TD B	2003/04/01 12:42
2	BRS	L2	7063	(page or web or webpage or web-page or (web adj page)) and (weight\$5 and scor\$4 and relevan\$5)	USPAT; US-PG PUB; EPO; JPO; DERW ENT; IBM_TD B	2003/04/01 13:06
3	BRS	L3	1466	2 and rank\$4	USPAT; US-PG PUB; EPO; JPO; DERW ENT; IBM_TD B	2003/04/01 12:43
4	BRS	L4	243	3 and (keyword oe key-word or (key adj word)) and (search\$5 adj engine)	USPAT; US-PG PUB; EPO; JPO; DERW ENT; IBM_TD B	2003/04/01 12:49
5	BRS	L5	1	4 and (convergenc\$4 near2 score)	USPAT; US-PG PUB; EPO; JPO; DERW ENT; IBM_TD B	2003/04/01 12:46

	Type	L #	ots	Search Text	DBs	Time Stamp
6	BRS	L6	4	4 and (convergenc\$4 same score)	USPAT; US-PG PUB; EPO; JPO; DERW ENT; IBM_TD B	2003/04/01 12:47
7	BRS	L8	1	7 and multi-keyword	USPAT; US-PG PUB; EPO; JPO; DERW ENT; IBM_TD B	2003/04/01 12:47
8	BRS	L9	1	4 and multi-keyword	USPAT; US-PG PUB; EPO; JPO; DERW ENT; IBM_TD B	2003/04/01 12:48
9	BRS	L10	1	3 and multi-keyword	USPAT; US-PG PUB; EPO; JPO; DERW ENT; IBM_TD B	2003/04/01 12:48
10	BRS	L11	85	3 and (pultiple or plurality or number) adj2 (keyword or key-word or (key adj word))	USPAT; US-PG PUB; EPO; JPO; DERW ENT; IBM_TD B	2003/04/01 12:49

	Type	L #	ots	Search Text	DBs	Time Stamp
11	BRS	L12	54	11 and (keyword or key-word or (key adj word)) and (search\$5 adj engine)	USPAT; US-PG PUB; EPO; JPO; DERW ENT; IBM_TD B	2003/04/01 12:49
12	BRS	L13	16	12 and rank\$4 near4 (page or web or webpage or web-page or (web adj page))	USPAT; US-PG PUB; EPO; JPO; DERW ENT; IBM_TD B	2003/04/01 12:52
13	BRS	L14	11	12 and rank\$4 near2 (page or web or webpage or web-page or (web adj page))	USPAT; US-PG PUB; EPO; JPO; DERW ENT; IBM_TD B	2003/04/01 13:09
14	BRS	L7	23	4 and (convergenc\$4)	USPAT; US-PG PUB; EPO; JPO; DERW ENT; IBM_TD B	2003/04/01 13:02
15	BRS	L15	4	7 and inbound and outbound	USPAT; US-PG PUB; EPO; JPO; DERW ENT; IBM_TD B	2003/04/01 13:03

	Type	L #	Reqs	Search Text	DBs	Time Stamp
16	BRS	L16	1	2 and (inbound and outbound) near2 link\$4	USPAT; US-PG PUB; EPO; JPO; DERW ENT; IBM_TD B	2003/04/01 13:06
17	BRS	L17	4	intrinsic adj rank\$4	USPAT; US-PG PUB; EPO; JPO; DERW ENT; IBM_TD B	2003/04/01 13:05
18	BRS	L18	3	17 and extrinsic adj rank\$4	USPAT; US-PG PUB; EPO; JPO; DERW ENT; IBM_TD B	2003/04/01 13:05
19	BRS	L19	131	(inbound and outbound) near2 link\$4	USPAT; US-PG PUB; EPO; JPO; DERW ENT; IBM_TD B	2003/04/01 13:07
20	BRS	L20	1	19 and (page or web or webpage or web-page or (web adj page)) and (weight\$5 and scor\$4 and relevan\$5)	USPAT; US-PG PUB; EPO; JPO; DERW ENT; IBM_TD B	2003/04/01 13:08

Patent Assignment Abstract of Title

Total Assignments: 1

Application #: 09757435 **Filing Dt:** 01/10/2001 **Patent #:** NONE **Issue Dt:**
PCT #: NONE **Publication #:** 20020129014 **Pub Dt:** 09/12/2002

Inventors: Brian S. Kim, Sudong Chung, Anurag Dod, Michael Kim, Yeogirl Yun

Title: Systems and methods of retrieving relevant information

Assignment: 1

Reel/Frame:	Received:	Recorded:	Mailed:	Pages:
<u>011525/0801</u>	02/23/2001	02/12/2001	05/02/2001	4

Conveyance: ASSIGNMENT OF ASSIGNORS INTEREST (SEE DOCUMENT FOR DETAILS).

Assignors: <u>KIM, BRIAN S.</u>	Exec Dt: 02/06/2001
<u>CHUNG, SUDONG</u>	Exec Dt: 02/06/2001
<u>DOD, ANURAG</u>	Exec Dt: 02/06/2001
<u>KIM, MICHAEL</u>	Exec Dt: 02/06/2001
<u>YUN, YEOGIRL</u>	Exec Dt: 02/06/2001

Assignee: WISENUT, INC. A CORPORATION OF THE STATE OF DELAWARE

SUITE 269
 1333 LAWRENCE EXPRESSWAY
 SANTA CLARA, CALIFORNIA 95051

Correspondent: ROBERT MOLL
 1173 SAINT CHARLES COURT
 LOS ALTOS, CA 94024

Search Results as of: 4/1/2003 8:44:13 A.M.

If you have any comments or questions concerning the data displayed, contact OPR / Assignments at 703-308-9723
 Web interface last modified: Oct. 5, 2002

DOCUMENT-IDENTIFIER: US 20020129014 A1

TITLE: Systems and methods of retrieving relevant information

----- KWIC -----

The present invention provides systems and methods of retrieving the pages according to the quality of the individual pages. The rank of a page for a keyword is a combination of intrinsic and extrinsic ranks. Intrinsic rank is the measure of the relevancy of a page to a given keyword as claimed by the author of the page while extrinsic rank is a measure of the relevancy of a page

on a given keyword as indicated by other pages. The former is obtained from the analysis of the keyword matching in various parts of the page while the latter is obtained from the context-sensitive connectivity analysis of the links connecting the entire Web. The present invention also provides the methods to solve the self-consistent equation satisfied by the page weights iteratively in a very efficient way. The ranking mechanism for multi-word query is also described. Finally, the present invention provides a method to obtain the more relevant page weights by dividing the entire hypertext pages into distinct number of groups.

[0002] The World Wide Web (Web) is a rapidly growing part of the Internet. One group estimates it grows roughly seven million Web pages (pages) each day

adding to an already enormous body of information. One study estimates there are more than two billion publicly available pages representing a growing fraction of the world's information. However, because of the Web's rapid growth and lack of central organization, millions of people cannot find specific information in an efficient manner.

[0003] To understand the problem, one must understand how the Internet and the Web are organized. The Internet is a communications infrastructure, which links computers throughout the world. It provides certain basic rules, termed protocols, by which computers can send data to each other. When a computer is

ready to send the data, it uses software to break data into packets that conform to the Internet Protocol (IP) and the Transmission Control Protocol (TCP). IP governs how packets of information are sent over the Internet. TCP allows one computer to send a stream of data to another by breaking the data

1-3
5-7
11-13
15-17
18-20
4, 8-10
14, 18-20

into packets, reassembling the packets at the receiving computer, and resending any missing packets. To do this they label each packet with a unique number and send it over the network. The receiving computer uses its Internet software to put the data in order. The data can be nearly anything: text, email, images, sounds, and software. The Web is the innovation of Tim Berners-Lee. See Berners-Lee, Weaving the Web (1999). His fundamental innovation was to provide a universal accessible **hypertext** medium for sharing information on the Internet. He understood that to become valuable the Web required many publishers. Because information constantly changes, it requires that any authorized person must be able to publish, correct, and read that information without any central control. Thus, there is no central computer governing the Web, and no single network or organization that runs it. To publish information, a person only needs access to a Web server, a computer program that shares Web resources with other computers. The person operating

the Web server defines who contributes, modifies and accesses the information. In turn, to access that information, a person only needs a client computer system, and a computer program, such as a browser, which can access the server

to read, edit, and at times correct the information displayed.

[0004] To be universally accessible, the Web is as unconstrained as possible. To allow computers to talk to each other everywhere, there are only a few basic rules: all resources on the Web, termed **Web pages** or pages, are identified by an address, termed a URL (Uniform Resource Locator). Once a page

has a URL, it can be published on a Web server and found by a browser. For example, one URL is <http://www.amazon.com/>. The letters to the left of the double slashes tell the browser what protocol to use, here HTTP, to look up the page. The part to the right--www.amazon.com identifies the Web server where the page exists. HTTP, a computer language, specifies which computer talks first and how to talk in turn. HTTP supports **hypertext**, nonsequential text, which **links** the pages together. Hidden behind a **hypertext** word, phrase, symbol, or image is the destination page's URL, which tells the browser where to locate the page. The loosely linked sets of pages constitute an information web. Once the computers agree to this conversation, they need a common language so they can understand each other. If they use the same software, they can proceed, otherwise they can translate to HTML (**Hypertext** Markup Language), a computer language supporting **hypertext**, and the language most persons currently use to write pages. It should be understood, however, that other languages such as XML, SMGL, as well as Java and JavaScript could be used to write pages.

[0005] In short, the Web is all information accessible to computers, where a URL identifies each unit of information. The Web has no central index to the pages, such as that contained in a public library. Instead, the pages have addresses and are loosely organized by links to each other. Thus, the Web provides little structure to support retrieval of specific information.

Instead, the Web creates a hypertext space in which any computer can link to any other computer.

[0006] Practically any computer can display pages through a browser such as Microsoft Internet Explorer or Netscape Communicator once connected to the Internet. Upon request the browser will fetch the page, interpret the text and display the page on the screen. The page may contain hypertext links, which are typically represented by text or an icon that is highlighted, underlined, and/or shown in a different color. The text or icon is referred to as anchor text. To follow the link, the user will move the cursor over the anchor text and click the mouse.

[0007] Several techniques exist for retrieving specific information. If the URL of the page is known, browsing the page suffices. If the Web site is known, one can go to the Web site map, search the site, or follow the links. This often works when the information is known to exist within a Web site. However, if the URL and site are unknown, finding information requires other techniques. Two known techniques are Web directories and search engines. For

example, Yahoo! classifies information in a hierarchy of subjects, such as Computer & Internet and Education. One chooses a category, then successive subcategories that seem likely to lead one to the information sought. But the categories are not mutually exclusive so multiple paths appear in the hierarchy. Once a category is selected, the previous category disappears forcing one to retrace one's steps to consider the other paths not taken. The further the search goes into the hierarchy, the more difficult it is to remember what other paths could be explored. To assist in searching the categories, Yahoo! provides phrase searching, and logical operators such as AND, OR, and NOT to specify which keywords must be present or absent in the pages, truncation of keywords, name searching, and field searching, e.g., in the title or URL.

[0011] Because thousands of pages may outwardly match the search criteria, the major search engines have a ranking function that will rank higher those pages having certain keywords in certain locations such as the title, or the Meta tag, or at the beginning of a page. This does not, however, typically put the most relevant page at the top of the list; much less assess the importance of the page relative to other pages. Moreover, relying solely on the content of the page itself--including the Meta tags which do not appear when

displayed--to rank the page can be a major problem to the search engine. A web author can repeat "hot" keywords many times, termed spamming, for example, in the title or Meta tags to raise the rank of a given page without adding value.

[0012] Unlike standard paper documents, the Web includes hypertext, which links one page to another and provides significant information through the link structure. For example, the inbound links to a page help to assess the importance of the page. Because some of the inbound links originate from authors other than the one who wrote the page being considered, they tend to give a more objective measure of the quality or importance of the pages. By making a link to other page, the author of the originating page endorses the destination page. Thus, to make your page highly regarded in this kind of ranking system, you need to convince a lot of other people to put links to your page in their pages.

[0013] Simple counting of inbound links, however, will not tell us the whole story. If a page has only one inbound link, but that link comes from a highly weighted page such as the Yahoo! home page, the page might be reasonably ranked higher than a page that has several inbound links coming from less visited pages.

[0014] The present invention relates to systems and methods of information retrieval. In one embodiment, a search engine and a method produces relevant search results to keyword queries. The search engine includes a crawler to gather the pages, indexer(s) to extract and index the URLs of pages with keywords into indexed data structure(s), and ranks hypertext pages based on intrinsic and extrinsic ranks of the pages based on content and connectivity analysis. The page weights can be calculated based on an iterative numerical procedure including a method for accelerating the convergence of the scores. The search engine can also rank pages based on scores of a multi-keyword query. The ranking scores can be based on using the entire set of hypertext pages and/or a subset based on topic or the like.

[0015] One embodiment of the present invention provides a general-purpose search engine a method to rank the pages without limitation to topic according to the quality of individual pages. This embodiment of the present invention enables the search engine to present the search results in such a way that the most relevant results appear on the top of the list.

[0017] FIGS. 2A-2C illustrate embodiments of the Web page database.

[0018] FIG. 3 illustrates the extrinsic rank and the intrinsic rank of a Web page.

[0020] FIG. 5 illustrates the page weight generator of the ranker.

[0022] The present invention relates to retrieving relevant pages from hypertext page collections. For conciseness, we describe a search engine for collecting, storing, indexing, and ranking Web pages in response to searcher queries. However, one of ordinary skill will understand after review of the specification that the search engine can be used on many other collections of hypertext pages.

[0023] FIG. 1 illustrates one embodiment of a search engine 10, which includes a crawler 12 to fetch pages from the Web 13. In one embodiment, the search engine 10 is written in C++, runs on the Microsoft 2000 Server operating system, preferably in parallel using suitable Intel Pentium processors. It should be clear, however, that it is not essential to the invention that this hardware and operating system be used, and other hardware and operating systems

can be used such as Solaris or Linux. Preferably, we run multiple instances of the crawler 12 to increase its capacity to crawl and re-crawl enormous hypertext document collections such as pages on the Web. The crawler 12 stores

the fetched pages in a Web page database 14, which includes data structures optimized for fast access of the fetched pages as explained below in connection with FIGS. 2A-2C.

[0024] The crawler 12 also sends the fetched pages to a link extractor 16, which finds the outbound links in the pages and sends the source and destination URLs of the links to a URL management system (UMS) 18. The UMS 18

assigns an identification number to each URL and maintains a database of identification number and URL pairs preferably in a hash table. The UMS 18 checks the URLs one by one, and if a new URL is found, it is sent to the crawler 12 to be written in the Web page database 14 preferably through a rate controller 20. The rate controller 20 buffers the URLs to be crawled, and sends the URLs to the crawler 12 only when the Web site providing the new page

has not received a crawling request for a certain amount of time. This ensures a Web site does not get excessive crawling requests.

[0025] The search engine 10 provides an indexing function in the following manner. The anchor text and link extractor 22 writes the source URL

identification number, the destination URL identification number, and associated anchor text to the anchor text and link database 24. Anchor text is a section of text, an icon, or other element in a page that links to another page. The indexer 26 extracts the anchor text from the anchor text and link database 24 and parses the keywords from the Web page database 14 and generates an indexed database 28. The indexer 26 stores each keyword and its associated list of URL identification numbers for fast retrieval.

[0026] The search engine 10 ranks the pages in the following manner. The ranker 30 reads the link structure from the anchor text and link database 24, calculates the page weight, reads the indexed words from the indexed database 28, and calculates the rank value for each keyword and page pair. The ranker

30 stores the page weight and the rank values in the indexed database 28.

The

ranker 30 also builds a ranked database 32 as a subset of the indexed database 28 for a single keyword query.

[0027] One purpose of the search engine is to respond to a search query with the search results in order of relevancy. The ranker 30 measures the relevancy

of each page by examining its content, its page weight, the weight of the links to the page, and the weight of the linking pages. The words, the font size, and the position of the words define the content of the page. For example, the ranker 30 compares the font size of each word relative to other words in the page to determine how important is the word. The position of the word in the page also may matter. For example, if the word is in the title, it may be desirable to give it more weight than if the word appears at the bottom of the page. Page weight is the probability of visitors viewing the page. The ranker 30 determines the intrinsic rank of the page by multiplying the content score by the page weight.

[0028] The ranker 30 measures the weight of the links to the page by examining the text associated with the link such as the anchor text. For example, the ranker 30 gives more credit to a link from a page with the keyword in its anchor text. The ranker 30 determines the extrinsic rank of the page by multiplying the link weight by the page weight.

[0029] When a query server 34 receives a query from a searcher 36, it collects the relevant pages from the ranked database 32 and indexed database 28. Given the huge size of these databases, the pages are stored across

multiple instances of search nodes. Each search node in the query server 34 collects a fixed number of the most relevant results from the databases it manages and returns them to query server 34. The query server 34 then sorts and ranks the results from these different nodes and presents the most relevant results, typically ten, at a time.

[0030] FIGS. 2A-2C illustrate that the data structures of the Web page database 14 typically need to accommodate different size pages. As shown in FIG. 2A, pages 1-4 and 6 show that most compressed pages are roughly the same

size, that is, not exceeding, for example, 6K bytes, while page 5 represents that some are larger. In FIG. 2B, the data structure stores pages adjacent to each other in a record file to save storage space, and provides an index file containing the starting address of each page. This requires the Web page database 14 engage in two steps for retrieving a page. In FIG. 2C, the preferred data structure leverages the fact that many of the compressed pages are below a threshold, e.g., 6K bytes, and stores those pages in a fixed record large enough to contain 85% of the pages. If the page is smaller, the fixed record has some amount of empty space. If the page is larger, the Web page database 14 stores as much of the page as possible in the fixed record and the rest in a record file. Thus, the Web page database 14 requires just one step to retrieve 85% of the pages, greatly reducing access time. One of ordinary skill would understand that it is neither essential to set the threshold at 85% of the pages, nor set the size of the fixed record to 6K bytes. To conserve storage space, a known compression technique such as the Microsoft Foundational

Class (MFC) zlib library can compress the pages by 4 to 1 or 3 to 1.

[0031] FIG. 3 illustrates how the ranker 30 determines the relevancy of a page by a combination of the intrinsic rank and extrinsic rank. Suppose we want to rank the relevancy of page a to a search query with keyword K.

[0032] We calculate intrinsic rank by looking at the content of page a. The author of page a can repeat the keyword K many times to claim page a is relevant to keyword K, but this relatively high content score can be adjusted by the weight of page a, that is, the importance of page a as indicated by other pages. Thus, the intrinsic rank of page a for keyword K is defined as the content score of page a multiplied by the page weight of the page a.

[0033] We calculate the extrinsic rank of page a by looking at the inbound links shown here as originating from page b and page c. Page b is shown as having one outbound link so its link weight is 1.0, while page c has two outbound links, which reduces each link weight to 0.5. The anchor weight for keyword K is obtained for each of these links. In one embodiment, anchor

weight is equal to link weight if the keyword K is found in the anchor text and zero if otherwise. In another embodiment, anchor weight can have smaller value than link weight if the keyword K is not found in the anchor text but related to or in the vicinity of the anchor text. The extrinsic rank is defined as the sum of the anchor weight multiplied by the page weight of all the pages shown here as page b and page c with inbound links to page a.

[0034] The overall rank of page a can be then calculated by combining the intrinsic rank and extrinsic rank in the following formula:

[0036] WR(K;a): Rank of page a for keyword K.

[0037] IR(K;a): Intrinsic rank of page a for keyword K.

[0038] ER(K;a): Extrinsic rank of page a for keyword K.

[0039] The adjustable parameter e determines the relative importance of the extrinsic rank with respect to the intrinsic rank. For instance, e=5 can be used but the value is not essential to the present invention.

[0040] Intrinsic Rank

[0041] Intrinsic rank is the measure of the importance of a page for a given keyword as claimed by the author of the page. Importance can be measured by examining the content of the page, e.g., the appearance of the keyword in the title or headings or body of the text, but this may be misleading. Some websites exaggerate the importance of their pages by repeating "hot" keywords endlessly without adding any value to the content of the page. We respect the author's claim as much as the page is worth. If the page is highly respected and frequently cited, we value author's claims more, and if otherwise, less. One solution is to use the page weight:

[0051] Page weight of a page is defined as the probability for a user--who travels on the Web endlessly in a random but well-defined manner--to visit the page. If a page has high probability to be visited by the user, the page is more likely to be a well-known page and to have many links from other pages.

[0052] In one embodiment, we can calculate the page weight by adding a hypothetical page, termed a page weight reservoir to the collection of pages. The page weight reservoir has a bi-directional link to each page in the collection. The page weight reservoir acts as a sink for pages having no outbound links (terminal pages) and a source for pages having no inbound links.

The page weight reservoir also solves the problem of pages pointing only to each other producing a loop, which would trap the user, and ensures the conservation of total page weight in the collection of pages.

[0053] The user complies with certain rules in moving from page to page. First, at each step, the user chooses an outbound link randomly and follows it to other pages. Second, if the user comes to the page weight reservoir, the user immediately chooses an outbound link randomly to the other pages. Consequently, each move from page to page is independent from prior history and it only depends on the current page.

[0054] Let $LW(a.fwdarw.b)$ denote the link weight, that is, the probability of choosing a particular outbound hyperlink to page b out of all outbound links originating from page a. The probability that the user visits page a at step n after visiting page b through the link $b.fwdarw.a$ is $LW(b.fwdarw.a).multidot.P.sub.n-1(b)$, where $P.sub.n-1(b)$ denotes the probability that the user visits page b at step n-1. Thus, we can write the probability of the user visiting page a at step n of the random walk, $P.sub.n(a)$, by collecting the contributions from all other pages as follows: $1 P n(a) = b LW(b a) P n - 1(b)$

[0062] Link Weight

[0063] Link weight is the probability for the user to choose a particular outbound hyperlink out of all outbound links originating from a page. Link weight also represents the importance of the link. In one embodiment, all link weights from a given page a can have a uniform value corresponding to $1/N.sub.out(a)$, where $N.sub.out(a)$ is the total number of links outbound from page a, including the extra link to the page weight reservoir. Therefore, $N.sub.out(a)$ is greater than or equal to one for every page and there is no terminal page in the collection. In another embodiment, not every outbound link is equally important. Thus, we give each link a different weight depending on several factors such as the offset of the link (i.e., position on the page) and the size of the paragraph where the link is located. A link readily visible upon the loading of a page can have a higher link weight than one visible only after scrolling down. The search engine can also assign different weights for external links--links that point to pages in other domain--and internal links--links that point to pages in the same domain. Many times the internal links serve as navigational tools rather than leading to new subjects represented by the anchor texts. The sum of all link weights from a page equals one: $5 b L W(a b) = 1$

[0064] If there is no link from one page to another, the corresponding link

weight is zero.

[0065] Extrinsic Rank

[0066] Extrinsic rank is a measure of the importance of a page for a given keyword as indicated by other pages. It measures the authoritativeness of a page on a given topic or keyword as regarded by public. Once the page weight is obtained the extrinsic rank can be calculated for each keyword and page pair. Extrinsic rank is defined as follows: $E R (K ; a) = b A W (K ; b a) P W (b)$

[0068] The equation multiplies the anchor weight of a link by the weight of the originating page and sums each product for all fetched pages. The anchor weight can be set in many different ways. The anchor text for a given link is useful for setting the anchor weight. We can also consider the related text of the page, which is either nearby the anchor text and/or related to the same topic. Thus, related headings, text in the vicinity of the anchor, and other anchor text on the same page may be useful for setting the anchor weight.

[0069] In one embodiment, $A W (K ; b . f w d a r w . a) = L W (b . f w d a r w . a)$ if the keyword is in the anchor text and zero if not. In another embodiment, we assign anchor weight less than or equal to the link weight if the keyword is found in text nearby or related to the anchor text. Thus, related headings, text in the vicinity of the anchor, and other anchor text on the same page can be used to set the anchor weight.

[0071] In another embodiment, the ranker 30 calculates the rank of pages for multi-keyword queries in the following manner. For simplicity, let's consider two-keyword query, K.sub.1,K.sub.2.

[0073] Where the intrinsic rank for the multi-keyword query is obtained as,

[0075] Where $P X (K . s u b . 1 , K . s u b . 2 ; a)$ is the proximity value between two keywords, K.sub.1 and K.sub.2, in page a. The proximity value has the maximum value of one when K.sub.2 immediately follows K.sub.1 and decreases to a minimum value as the distance between two keywords grows. The proximity value decreases to its minimum, such as 0.1, when two keywords are separated by more than 10 words, and K.sub.1 K.sub.2 represents the Boolean AND operation of the

two keywords. For computing the extrinsic rank, $ER(K.\text{sub.1}, K.\text{sub.2}; a)$, we need to also introduce the concept of partial extrinsic rank as explained in the following section.

[0076] The partial extrinsic rank is defined as: $7 PER(UA; a) = c AW(UA; ca) PW(c)$

[0077] Where page c represents all pages, which contains link to page a with the identical anchor text, UA . In other words, we collect the contributions to extrinsic rank from all pages with identical anchor text into one partial extrinsic rank, which saves computational resources when calculating proximity value. Thus, the partial extrinsic rank is very useful for a multi-keyword query. In another embodiment, partial extrinsic rank can be used for a single keyword query and will be the sum of partial extrinsic ranks over the identical anchor text: $8 ER(K; a) = UA(K) PER(UA(K); a)$

[0079] In one embodiment, the ranker 30 uses the partial extrinsic rank to obtain the extrinsic rank for a multi-keyword query in the following manner: $9 ER(K_1, K_2; a) = UA(K_1, K_2) PER(UA(K_1, K_2); a) PX(K_1, K_2; UA(K_1, K_2))$

[0080] $UA(K.\text{sub.1}, K.\text{sub.2})$ is the identical anchor text containing both keywords $K.\text{sub.1}$ and $K.\text{sub.2}$. $PX(K.\text{sub.1}, K.\text{sub.2}; UA(K.\text{sub.1}, K.\text{sub.2}))$ is the proximity value of the keywords $K.\text{sub.1}$ and $K.\text{sub.2}$ within the identical anchor text $UA(K.\text{sub.1}, K.\text{sub.2})$. To facilitate the extrinsic rank calculation of multi-keyword query, the indexed database 28 contains a field to store the partial extrinsic rank for each identical anchor text and stores all offset for each keyword in the anchor text. Therefore, to calculate the extrinsic rank for multi-word query, we can find the entry for $K.\text{sub.1}$ and $K.\text{sub.2}$ in indexed database 28. For each page there will be a list of identical anchor text with its identification number, the offset of the keyword in the anchor text, and the partial extrinsic rank. From the identical anchor text identification number and offset, the ranker 30 can obtain the proximity value and it collects the product of partial extrinsic rank and proximity value.

[0083] The numbers in the table will be used for the anchor weight. Using these tables, when we calculate the extrinsic rank for "automobile", for example, we can collect the keyword "car" at the same time. Further, the anchor text containing "truck" contributes, but with less weight.

[0084] As illustrated in FIG. 4, the ranker 30 includes a link structure sorter 38, which extracts records of the link structure from the anchor text and link database 24, and sorts the records in the order of source URL identification number. The sorted records are written in the link database 40.

The page weight generator 42 reads the link database 40 and calculates the page weight for the fetched pages and stores them in the page weight database 43. The intrinsic rank generator 44 reads the indexed database 28 to calculate the content score and multiplies it with the page weight read from the page weight database 43 to calculate the intrinsic rank for a given keyword and URL pair. The intrinsic rank generator 44 reads one keyword at a time from the indexed database 28. The indexed database 28 stores a set of records where each record includes the URL identification number and bit fields to indicate the presence and proximity of a given keyword in the title, the anchor text of the inbound link text related to the anchor text as described earlier, in the plain text, and/or in the URL of the page. Another bit field of the record can be set when the URL is the top-level of a given host.

[0085] The partial extrinsic rank generator 47 reads several input files: the anchor text and link database 24, the indexed database 28, and the page weight database 43, and calculates the partial extrinsic rank values for each identical anchor text and URL pair. The resulting partial extrinsic rank is written back to the indexed database 28 to be used for extrinsic rank for single and multi-word query.

[0086] The extrinsic rank generator 45 collects the partial extrinsic rank for each keyword and URL pair. In the case of a multi-keyword query, the extrinsic rank generator 45 collects all partial extrinsic ranks for identical anchor text containing the keywords.

[0087] Intrinsic and extrinsic rank values are sent to the merger module 46 to be combined into the final rank. The merger module 46 updates the rank of each URL in the indexed database 28. The merger module 46 collects the top-ranked URLs (e.g., top 400 URLs) and writes them in the ranked database 32 in descending order.

[0088] FIG. 5 illustrates the page weight generator 42 of the ranker 30. The function 50 initializes the page weight vector X to a constant such as 1. The connectivity graph G 48, representing the link structure of all of the fetched pages, is constructed from the link database 40. The function 52 takes the connectivity graph G 48 and the new input page weight vector X 54 and computes the output page weight vector Y. The function 56 tests for convergence. If the output page weight vector Y is satisfactorily close to the input page weight vector X within a predetermined tolerance, typically in the order of 10^{-6} , then the iteration stops and the final page weight vector is written to the page weight database 43. If the convergence is not achieved,

the function 56 passes the input and output page weight vectors, X and Y, to the mixer module 58, which mixes them to generate the new input page weight vector X 54, and the iterative process repeats until the convergence is reached. In function 56, we use a normalized error function to measure the convergence as follows: $10 e = \sum_i (y_i - x_i)^2 / \sum_i (x_i)^2$

[0092] X is a N-1 column matrix representing the page weights for all N fetched pages. N.times.N square matrix G represents the connectivity graph. The off-diagonal elements of G represent the link connectivity between the pages. The diagonal elements of the matrix G are all equal to zero. The solution vector X is an eigenvector of the matrix G with the eigenvalue one. In principle, the solution vector X can be obtained from solving this matrix equation exactly. In dealing with the World Wide Web, however, the number of total pages N is very large--order of hundred of millions or billions--and solving this matrix equation exactly is impractical in terms of computer memory and CPU time. Thus, we employ the iterative method. We start with a guess for X in the right-hand-side to obtain X in the left-hand-side. In general, the input and output X will not be same and we combine the input and output X to prepare new input X and iterate this process until they become self-consistent within the preset tolerance.

[0098] In another embodiment, the present invention provides a method for calculating ranking scores by dividing the entire hypertext pages into distinct number of groups. The page weight can be refined further by classifying pages into several distinct groups such as "Art", "Education", "Reference", etc. Once the pages are classified, the page weights can be calculated in the same manner described above within these groups. The page weight will be more relevant for the given topic.

1. A computer-implemented method of ranking the relevancy of a collection of hypertext pages to a keyword-based query, comprising: calculating an intrinsic rank of a page; calculating an extrinsic rank of the page; and calculating the rank of the page by combining the intrinsic rank and the extrinsic rank.
2. The method of claim 1, wherein the intrinsic rank is a function of the content score and the page weight of the page.
4. The method of claim 2, wherein the page weight is defined as the probability of a user visiting the page when traveling in the collection of hypertext pages in a random fashion.
5. The method of claim 2, wherein the page weight is obtained as the sum of the product of a link weight of each inbound link to the page and the page

weight of the originating page.

6. The method of claim 2, wherein the page weight is computed by the following steps of: constructing a connectivity graph, which represents the collection of hypertext pages and the link structure between the pages; adding a page weight reservoir with bidirectional links to and from each of the pages in the collection of hypertext pages; and summing all of the products of each inbound link weight with the page weight of the originating page providing the inbound link.

7. The method of claim 2, further comprising computing the page weights by the following steps of: initializing a page weight vector to a constant; constructing a connectivity graph representative of the link structure of the collection of pages; computing an output page weight vector from the input page weight vector and the connectivity graph; and comparing the output page weight vector with the input page weight vector for convergence, and if convergence is reached, writing the output page weight vector in a page weight database, and if not, mixing the input and output page weight vectors to generate a new input page weight vector and repeating until convergence is reached.

8. The method of claim 5, wherein the link weight is defined as the probability of a user randomly choosing the link to visit other pages when traveling in the collection of hypertext pages.

9. The method of claim 5, wherein the link weight of the inbound links has a uniform value corresponding to the reciprocal of the total number of links outbound from an originating page.

10. The method of claim 5, wherein the link weight has a variable value, which depends on the number of outbound links, the offset of the link, the size of the paragraph where the link is located, and/or whether the link is an external or internal link.

11. The method of claim 1, wherein the extrinsic rank is a function of the anchor weight and the page weight of the pages providing inbound links to the page.

12. The method of claim 1, wherein the extrinsic rank is obtained by summing the products of the anchor weight and the page weight of the originating page providing each inbound link.

13. The method of claim 11, wherein the anchor weight is a function of the inbound link weights and the keyword being present in the anchor text, in the

vicinity of the anchor text, or in text related to the topic of the anchor text.

14. The method of claim 11, wherein the page weight is defined as the probability of a user randomly visiting a page in the collection of hypertext pages.

15. The method of claim 11, wherein the page weight is obtained by summing the products of the link weight of each inbound link to the page and the page weight of the originating page providing the inbound links.

16. The method of claim 11, wherein the page weight is computed by the following steps of: constructing a connectivity graph, which represents the collection of hypertext pages and the link structure between the pages; adding a page weight reservoir with bi-directional links to and from each of the pages in the collection of hypertext pages; and summing all of the products of each inbound link weight with the page weight of the originating page providing the inbound link.

17. The method of claim 11, further comprising computing the page weights by the following steps of: initializing a page weight vector to a constant; constructing a connectivity graph representative of the link structure of the collection of pages; computing an output page weight vector from the input page weight vector and the connectivity graph; and comparing the output page weight vector with the input page weight vector for convergence, and if convergence is reached, writing the output page weight vector in a page weight database, and if not, mixing the input and output page weight vectors to generate a new input page weight vector and repeating until convergence is reached.

18. The method of claim 15, wherein the link weight is defined as the probability of a user randomly choosing the link to visit other pages when traveling in the collection of hypertext pages.

19. The method of claim 15, wherein the link weight of the inbound links has a uniform value corresponding to the reciprocal of the total number of links outbound from an originating page.

20. The method of claim 15, wherein the link weight has a variable value, which depends on the number of outbound links, the offset of the link, the size of the paragraph where the link is located, and/or whether the link is an external or internal link.

21. The method of claim 1, wherein the collection of hypertext pages is

fetched from the Web.

22. A computer-implemented method of ranking a collection of hypertext pages, comprising: calculating the intrinsic rank of a page for a multi-keyword query; calculating the extrinsic rank of the page for the multi-keyword query; and calculating the rank of the page in the collection of hypertext pages by combining the intrinsic rank and the extrinsic rank.

23. The method of claim 22, wherein the intrinsic rank is a function of content score and the page weight.

25. The method of claim 22, wherein the extrinsic rank of the page is a function of the partial extrinsic ranks and the proximity value of the multi-keywords.

26. The method of claim 25, wherein partial extrinsic rank is a function of the anchor weight and the page weight of the pages with identical anchor text.

27. The method of claim 25, wherein partial extrinsic rank is computed by summing the products of the anchor weight and the page weight of the pages with identical anchor text.

28. A Web search engine, comprising: a Web page database; a crawler to fetch pages from the Web and store the pages in the Web page database; a link extractor to extract link information from the pages; a URL management system to assign an identification number to the URL of each page, and store the identification number and URL pairs in the Web page database and send new URLs to the crawler to be retrieved from the Web; anchor text and link database; an anchor text and link extractor to extract the anchor text and the link information from the pages and store in the anchor text and link database; indexed database; an indexer to parse keywords from the pages and store the keyword and URL identification pairs in the indexed database; and a ranker to rank a page based on intrinsic rank and extrinsic rank of the page.

29. The Web search engine of claim 28, wherein the ranker determines the intrinsic rank from content information in the indexed database and the page

weight computed from the link information in anchor text and link database, and the extrinsic rank from the anchor text information in the anchor text and link database and the computed page weight.

30. The Web search engine of claim 28, wherein the ranker determines the intrinsic rank of the page based on the content score and the page weight.

31. The Web search engine of claim 28, wherein the ranker determines the extrinsic rank of the page based on the anchor weight of each inbound link and the page weight of the originating page.

32. The Web search engine of claim 28, wherein the ranker determines the anchor weight based on the link weight and the keyword being present in the anchor text or related text.

33. The Web search engine of claim 28, wherein the ranker calculates the intrinsic rank and extrinsic rank of a page for a multi-keyword query, wherein the intrinsic rank is a function of content score and the page weight, the extrinsic rank of the page is a function of the partial extrinsic ranks and proximity values.

34. The Web search engine of claim 28, further comprising a page weight generator and a page weight database, computing page weights by initializing a page weight vector to a constant, constructing a connectivity graph representing the link structure of the fetched pages, computing an output page weight vector from the input page weight vector and the connectivity graph, and comparing the output page weight vector with the input page weight vector and if convergence is reached, writing the output page weight vector in a page weight database, and if not, mixing the input and output page weight vectors to generate a new input page weight vector and repeating until convergence is reached.

35. A computer system for ranking search results from a query on a collection of hypertext pages, comprising: a crawler to fetch pages from the collection of hypertext pages; a link extractor to extract page locator information from the fetched pages; a page locator management system for storing and retrieving the page locator information; a page database to store the pages; an indexer to parse keywords from the pages and store the keyword page locator pairs in the indexed database; an anchor text and link extractor to extract the anchor text and link structures from the pages; an anchor text and link database, wherein the anchor text and link extractor writes the anchor text and link structures into the anchor text and link database; and a ranker to assign a rank value to a page based on intrinsic and extrinsic rank.

36. The system of claim 35, wherein the ranker assigns an intrinsic rank to

the page based on a combination of content score and page weight.

37. The system of claim 35, wherein the ranker assigns the content score to

the page for a keyword based on a combination of location, frequency, and/or font size of the keyword in the page.

38. The system of claim 35, wherein the ranker assigns a page weight to the page as the probability of a searcher visiting the page when traveling in the collection of hypertext pages in a random fashion.

39. The system of claim 35, wherein the ranker assigns a uniform value corresponding to the reciprocal of the total number of links outbound from an originating page to link weight.

40. The system of claim 35, wherein the ranker assigns link weight based on location of the link.

41. The system of claim 35, wherein the ranker assigns an extrinsic rank to
the page for a given keyword as a combination of anchor weight of the links from other pages and the page weight of referring pages.

42. The system of claim 35, wherein the ranker assigns a rank value to a
page for a multi-keyword query as a combination of intrinsic rank and
extrinsic
rank for the multi-keyword.

43. The system of claim 35, wherein the ranker assigns an intrinsic rank to
a page for a multi-keyword query as a combination of content score and page weight.

44. The system of claim 35, wherein the ranker assigns a content score to
a
page for a multi-keyword query as a combination of content score based on intersection of the given keywords and proximity value.

45. The system of claim 35, wherein the ranker assigns a partial extrinsic rank for each variation of identical anchor text.

46. The system of claim 35, wherein the ranker assigns a extrinsic rank to
a page for a multi-keyword query as a combination of partial extrinsic rank of identical anchor text and proximity values in each anchor text.

47. The system of claim 35, wherein the ranker obtains a link connectivity

graph of the pages.

47. The system of claim 35, wherein the ranker obtains the rank values from the link connectivity graph.

48. The system of claim 35, wherein the ranker calculates the page weight by iterative numerical procedure.

50. The system of claim 35, wherein the ranker calculates rank values by dividing the pages into distinct number of groups.

52. The system of claim 35, wherein the Web page database stores the pages

in a fixed record large enough to contain a predetermined percentage of all of the pages, wherein if the page is smaller, the fixed record has some empty space, and if the page is larger, the Web page database stores as much of the page as possible in the fixed record and the rest in a record file.

Patent Assignment Abstract of Title

Total Assignments: 1

Application #: 09561200 **Filing Dt:** 04/27/2000 **Patent #:** NONE **Issue Dt:**

PCT #: NONE

Publication #: NONE

Pub Dt:

Inventors: Yeogirl Yun, Steve Hyuntai Jung, Sudong Chung

Title: Keyword-based search engine system and method of retrieval

Assignment: 1

Reel/Frame:	Received:	Recorded:	Mailed:	Pages:
<u>010770/0496</u>	05/12/2000	04/27/2000	10/18/2001	3

Conveyance: ASSIGNMENT OF ASSIGNORS INTEREST (SEE DOCUMENT FOR DETAILS).

Assignors: YUN, YEOGIRL

Exec Dt: 04/27/2000

JUNG, STEVE HYUNTAI

Exec Dt: 04/27/2000

CHUNG, SUDONG

Exec Dt: 04/27/2000

Assignee: WISENUT, INC.

1333 LAWRENCE EXPRESSWAY
SANTA CLARA, CALIFORNIA 95051

Correspondent: ROBERT MOLL

@PATENTPLANET
1173 ST. CHARLES COURT
LOS ALTOS, CA 94024

Search Results as of: 4/1/2003 8:45:14 A.M.

If you have any comments or questions concerning the data displayed, contact OPR / Assignments at 703-308-9723
Web interface last modified: Oct. 5, 2002

	Type	L #	Units	Search Text	DBs	Time Stamp
21	BRS	L21	131	(inbound and outbound) near5 (link\$4 or hypertext or hyperlink\$4)	USPAT; US-PG PUB; EPO; JPO; DERW ENT; IBM_TD B	2003/04/01 13:08
22	BRS	L22	1	21 and (page or web or webpage or web-page or (web adj page)) and (weight\$5 and scor\$4 and relevan\$5)	USPAT; US-PG PUB; EPO; JPO; DERW ENT; IBM_TD B	2003/04/01 13:08
23	BRS	L23	131	(inbound and outbound) near (link\$4 or hypertext or hyperlink\$4)	USPAT; US-PG PUB; EPO; JPO; DERW ENT; IBM_TD B	2003/04/01 13:09
24	BRS	L24	131	(inbound and outbound) near (link\$4 or hypertext or hyperlink\$4)	USPAT; US-PG PUB; EPO; JPO; DERW ENT; IBM_TD B	2003/04/01 13:09
25	BRS	L25	1	24 and rank\$4 near2 (page or web or webpage or web-page or (web adj page))	USPAT; US-PG PUB; EPO; JPO; DERW ENT; IBM_TD B	2003/04/01 13:09

	Type	L #	Hits	Search Text	DBs	Time Stamp
1	BRS	L2	196	(rank\$3 near4 (page or web or webpage or web-page or (web adj page))) and (search\$5 adj engine)	USPAT; US-PG PUB; EPO; JPO; DERW ENT; IBM_TD B	2003/04/01 11:49
2	BRS	L3	125	2 and (keyword or key-word or (key adj word))	USPAT; US-PG PUB; EPO; JPO; DERW ENT; IBM_TD B	2003/04/01 11:46
3	BRS	L4	50	3 and weight\$5 and scor\$4 and relevan\$5	USPAT; US-PG PUB; EPO; JPO; DERW ENT; IBM_TD B	2003/04/01 11:48
4	BRS	L5	4	4 and (page or web or webpage or web-page or (web adj page)) near4 weight\$5 and scor\$4 and relevan\$5	USPAT; US-PG PUB; EPO; JPO; DERW ENT; IBM_TD B	2003/04/01 11:50

	Type	L #	Hits	Search Text	DBs	Time Stamp
1	BRS	L1	451	rank\$4 near6 page	USPAT; US-PG PUB; EPO; JPO; DERW ENT; IBM_TD B	2003/04/01 08:54
2	BRS	L2	206	1 and (hypertext or link or hyperlink) and (webpage or web-page or (web adj page))	USPAT; US-PG PUB; EPO; JPO; DERW ENT; IBM_TD B	2003/04/01 08:55
3	BRS	L3	0	1 and ((hypertext or link or hyperlink) and (webpage or web-page or (web adj page))) with relevan\$5c	USPAT; US-PG PUB; EPO; JPO; DERW ENT; IBM_TD B	2003/04/01 08:47
4	BRS	L4	33	1 and ((hypertext or link or hyperlink) and (webpage or web-page or (web adj page))) with relevan\$5	USPAT; US-PG PUB; EPO; JPO; DERW ENT; IBM_TD B	2003/04/01 08:51
5	BRS	L5	4	2 and intrinsic and extrinsic	USPAT; US-PG PUB; EPO; JPO; DERW ENT; IBM_TD B	2003/04/01 08:54

	Type	L #	Hits	Search Text	DBs	Time Stamp
6	BRS	L6	5265	intrinsic and extrinsic	USPAT; US-PG PUB; EPO; JPO; DERW ENT; IBM_TD B	2003/04/01 08:57
7	BRS	L7	4	6 and rank\$4 near6 page	USPAT; US-PG PUB; EPO; JPO; DERW ENT; IBM_TD B	2003/04/01 08:54
8	BRS	L8	245	6 and rank\$4	USPAT; US-PG PUB; EPO; JPO; DERW ENT; IBM_TD B	2003/04/01 08:56
9	BRS	L9	8	8 and (hypertext or link or hyperlink) and (webpage or web-page or (web adj page))	USPAT; US-PG PUB; EPO; JPO; DERW ENT; IBM_TD B	2003/04/01 08:57
10	BRS	L10	8	6 and (intrinsic and extrinsic) same rank\$4	USPAT; US-PG PUB; EPO; JPO; DERW ENT; IBM_TD B	2003/04/01 08:57

	Type	L #	Hits	Search Text	DBs	Time Stamp
11	BRS	L11	5	10 and (webpage or web-page or (web adj page) or page)	USPAT; US-PG; PUB; EPO; JPO; DERW; ENT; IBM_TD; B	2003/04/01 08:58